



Shao, C. (2020). Quantum speedup of Bayes' classifiers. *Journal of Physics A*, 53, [045301]. <https://doi.org/10.1088/1751-8121/ab5d77>

Peer reviewed version

Link to published version (if available):  
[10.1088/1751-8121/ab5d77](https://doi.org/10.1088/1751-8121/ab5d77)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IOP Publishing at <https://iopscience.iop.org/article/10.1088/1751-8121/ab5d77>. Please refer to any applicable terms of use of the publisher.

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

PAPER

## Quantum speedup of Bayes' classifiers

To cite this article: Changpeng Shao 2020 *J. Phys. A: Math. Theor.* **53** 045301

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices  
to create your essential collection of books in STEM research.

Start exploring the **collection** - download the first chapter of  
every title for free.

# Quantum speedup of Bayes' classifiers

Changpeng Shao 

School of Mathematics, University of Bristol, Bristol, BS8 1UG, United Kingdom

E-mail: [changpeng.shao@bristol.ac.uk](mailto:changpeng.shao@bristol.ac.uk)

Received 26 July 2019, revised 12 November 2019

Accepted for publication 29 November 2019

Published 3 January 2020



CrossMark

## Abstract

Data classification is a fundamental problem in machine learning. We study quantum speedup of the supervised data classification algorithms (quadratic, linear and naïve Bayes classifiers) based on Bayes' theory. The main technique we use to achieve quantum speedup is block-encoding. However, to apply this technique effectively, we propose a general method to construct the block-encoding. As an application, we show that all the three classifiers achieve exponential speedup at the number of samples over their classical counterparts. As for the dimension of the space, quantum quadratic and linear classifiers achieve varying degrees of polynomial speedup, while quantum naïve Bayes' classifier achieves an exponential speedup. The only assumption we make is the qRAM to prepare quantum states of the input data.

Keywords: quantum algorithms, quantum computing, Bayes' classifiers, machine learning

## 1. Introduction

In machine learning, data classification is a problem of identifying to which of a set of class a new data belongs, on the basis of a set of training samples whose class membership is known. Many effective algorithms to accomplish the task of classification were discovered in the past decades, such as support vector machines, neural networks,  $k$ -means,  $k$ -nearest neighbour, etc. They have wide applications in computer vision, drug discovery and development, speech recognition, pattern recognition and so on [1–4]. However, these methods become time-consuming when the number of training samples or the dimension of the space is large. Quantum computer is a new computing device that is good at manipulating high dimensional data. Consequently, the corresponding quantum classifiers should be more efficient than their classical counterparts.

For instance, in [5], Lloyd *et al* proposed exponential fast quantum algorithms to implement the nearest mean and  $k$ -means methods. In [6], Wiebe *et al* studied the quantum version of the nearest-neighbour method. Quadratic speedup at the number of training samples was obtained. Rebentrost *et al* in [7] obtained an exponential speedup to implement the least-squares support

vector machines in the quantum computer. In [8], Aïmeur *et al* showed that when having an oracle to compute the distance between two vectors, quadratic speedup of  $k$ -median clustering algorithm can be achieved in a quantum computer by Grover's searching algorithm. Under some oracle assumptions, Cong and Duan in [9] presented quantum algorithms to simulate Fisher discriminant analysis, which achieves exponential speedup at the number and the dimension of training samples. Quantum neural networks [10] were also studied with varying degrees of success. For example, in [11] Farhi and Neven applied variational quantum circuits to build quantum neural networks which might be a possible application of near-term quantum computer. Kernel-learning methods are ubiquitous for pattern classification and recognition. Recently, the connections between kernel methods and quantum computing have been studied in [12, 13]. This provides new approaches to design supervised learning algorithms in a quantum computer. For more about quantum machine learning, we refer to [14, 15].

The above mentioned supervised classification algorithms belong to the non-parametric methods, i.e. no statistical model assumptions on the training samples are made. In this paper, we will study quantum speedup of classification algorithms based on parametric method—Bayesian theory. Bayesian methods for machine learning is a field of interest due to its wide applications in text categorization, drug discovery, medical diagnosis, etc [16–18]. They enhanced many machine learning algorithms in handling missing data, extracting information from small datasets, and so on. Furthermore, Bayesian methods allow us to estimate the uncertainty in predictions. This is a desirable feature for fields like medicine.

In the parametric method [19, 20], we assume there is a statistical model that is valid over the whole input space. Then based on Bayes' theorem, the classification is performed by calculating the largest posterior probability, which is further reduced to calculate certain discriminant functions. The advantage of the parametric method is that it reduces the problem of estimating the discriminant functions to estimating a small number of parameters. Thus it can alleviate problems stemming from the curse of dimensionality in other classification algorithms.

The estimation of the discriminant functions relies on the computation of matrix inversion or the inner product of vectors. In the quantum computer, several efficient algorithms to compute the inverse of matrices were discovered in the past, such as HHL [21], singular value estimation (SVE) [22], block-encoding [23], etc. They have inspired many applications in quantum machine learning [5, 7, 9, 24–29]. Especially, in [28, 29], Zhao *et al* applied the HHL algorithm to accelerate the Bayesian training of deep neural networks in the quantum computer and the Gaussian process regression—a key component of the model-building at the core of Bayesian optimization. In comparison, inner product evaluation is much easier than matrix inversion. Some quantum algorithms are only based on inner product estimation. One interesting such algorithm [35], which is implemented in the IBM Quantum Experience, uses distance as the criterion to do binary pattern classification. The supervised clustering algorithm proposed in [5] also only depends on inner product evaluation.

Different from [28, 29], although this paper studies the quantum classification algorithms based on Bayesian theory, the HHL algorithm may not help to achieve speedup in computing the inverse of matrices. One obstacle to use HHL algorithm is the Hamiltonian simulation. Firstly, if the matrix is sparse, then the Hamiltonian simulation is efficient [30]. Unfortunately, this is not the case for the Bayes' classifiers. Secondly, the matrices in the discriminant functions are given indirectly. Some expensive calculations are required to obtain all the entries of the matrices, thus the methods proposed in [31, 32] to simulate dense Hamiltonian may not be applicable.

On the other hand, the SVE can be used to compute the inverse of dense matrices; nevertheless, it needs a binary tree data structure to store the matrices in the quantum computer in advance. Since the matrices studied in this paper are not given directly, it is not easy to build such data structures for them.

Fortunately, we can apply block-encoding to achieve quantum speedup in computing the matrix inversion. To apply the block-encoding method, in section 2 we first propose a general method to construct the block-encoding of Hermitian matrices. Roughly, it can be viewed as a generalization of SVE. However, it does not depend on the data structure of the given matrix. Instead, it only needs a unitary operator, which is usually much easier to construct, to prepare some quantum states that contain the quantum information of rows of the matrix. Compared to the original SVE, the modified one only achieves quadratic speedup at the dimension. But it may have a wider range of applications than the original one because of its flexibility.

Another technique we will utilize is the swap test [33], which can be used to estimate the inner product of two quantum states. It forms a basic operation of quantum states and has many applications in designing quantum algorithms [34–36]. We will discuss this in section 3.

Based on the different choices of the estimation of the statistics, there are at least four types of classifiers (see a brief review in section 4): quadratic, linear, naïve Bayes and nearest mean. The nearest mean has been studied in [5]. As for the other three, based on the block-encoding and swap test techniques, we will show in section 5 that all their quantum versions can achieve exponential speedup at the number of training samples over the classical counterparts. As for the dimension  $d$  of the space, quantum quadratic classifier achieves a minor polynomial speedup over the classical one ( $d^{2.5}$  versus  $d^3$ ), quantum linear classifier achieves a sextic speedup ( $d^{0.5}$  versus  $d^3$ ), and quantum naïve Bayes classifier achieves an exponential speedup ( $\log d$  versus  $d$ ).

This paper is organized as follows: in section 2, we introduce a general method to construct block-encoding and show two applications of it that are useful in this paper. In section 3, we give a general version of swap test that is useful in this work. The results in these two sections may have other applications in quantum machine learning. Section 4 aims to provide a simple review of Bayes' classifiers. Finally, in section 5 we present three quantum algorithms to accelerate the Bayes' classifiers.

## 2. Block encoding

When designing quantum algorithms in the standard quantum circuit model, we appropriately choose unitaries to operate on quantum states to increase the probability of the target states. Block-encoding [23, 37] is a method to encode non-unitary matrices as a sub-block of unitary operators so that all matrices can be used in designing quantum algorithms to certain degree. It was first introduced by Low and Chuang [38] in 2016 to perform optimal Hamiltonian simulation. A systemic study was given in [23], in which more efficient quantum algorithms to solve the least squares problem and estimate the electrical-network quantities were discovered based on block-encoding. Later, block-encoding has inspired many applications in machine learning, such as quantum clustering and classification algorithm [24, 25], quantum SDP solver [39, 40], quantum row and column iterative algorithm [41], etc.

In this section, we only consider Hermitian matrices. We first establish a method to find the block-encoding of Hermitian matrices. Then we focus on two specific problems that are useful in section 5: the estimation of the inner product  $\langle x|A^{-1}|y\rangle$  and the computation of the logarithm of the determinant  $\log |A|$ , where  $A$  is a Hermitian matrix.

Let  $A = (a_{ij})_{M \times M}$  be an  $M \times M$  Hermitian matrix. For convenience, assume that  $M = 2^m$ . Otherwise, we extend  $A$  into  $\text{diag}\{A, I\}$  by adding an identity matrix  $I$  of certain dimension. Denote the  $i$ th row of  $A$  as  $A_i$ , and the 2-norm of  $A_i$  as  $\|A_i\|_2$ . Assume that there is a unitary  $U_{\mathcal{M}}$  which runs in time  $O(T)$  and a constant  $\alpha$  such that

$$U_{\mathcal{M}}|i\rangle|0\rangle^{\otimes(m+1)} = |i\rangle\left(\alpha\|A_i\|_2|0\rangle|A_i\rangle + |1\rangle|G_i\rangle\right), \quad (1)$$

where  $|1\rangle|G_i\rangle$  refers to the quantum state orthogonal to the first term. For simplicity, we will write  $|1\rangle|G_i\rangle$  as  $|1\rangle(\dots)$  or just  $|0\rangle^\perp$  in this paper since usually we do not concern about the explicit form of  $|G_i\rangle$ . Moreover, the qubit  $|1\rangle$  or the notation  $|0\rangle^\perp$  is already enough to express that this state is orthogonal to the first term. If the number of qubit  $|0\rangle$  is not important, we will briefly denote  $|0\rangle^{\otimes(m+1)}$  as  $|0\dots 0\rangle$  or  $|0\rangle$ .

**Remark 1.**

(i). For each  $i$ , if there is an unitary operator  $U_i$  such that

$$U_i|0\rangle^{\otimes(m+1)} = \alpha\|A_i\|_2|0\rangle|A_i\rangle + |1\rangle|G_i\rangle, \quad (2)$$

then  $U_{\mathcal{M}} = \sum_{i=0}^{M-1} |i\rangle\langle i| \otimes U_i$  satisfies the condition (1). This is one simple way to construct  $U_{\mathcal{M}}$ . However, the implementation complexity of this kind of construction is usually affected by  $M$ . For the problems considered in this paper, we can construct  $U_{\mathcal{M}}$  more efficiently.

(ii). In this section, without other statements,  $O(T)$  always refers to the complexity to implement the unitary operator defined in equation (1), and  $\alpha$  is the constant used in equation (1).

**Definition 1 (Block-encoding [23]).** Suppose that  $A$  is an  $m$ -qubit operator,  $\gamma, \epsilon \in \mathbb{R}^+$  and  $k \in \mathbb{N}$ . Then we say that the  $(m+k)$ -qubit unitary  $U$  is an  $(\gamma, k, \epsilon)$ -block-encoding of  $A$ , if

$$\|A - \gamma(|0\rangle^{\otimes k} \otimes I_{2^m})U(|0\rangle^{\otimes k} \otimes I_{2^m})\| \leq \epsilon, \quad (3)$$

where  $\|\cdot\|$  is the spectral norm (i.e. the maximal singular value). Therefore,

$$U = \begin{pmatrix} A' & \cdot \\ \cdot & \cdot \end{pmatrix} \quad (4)$$

satisfies  $\|A - \gamma A'\| \leq \epsilon$ .

**Remark 2.** In block-encoding,  $A$  does not need to stay in the top-left corner of  $U$ . So for instance

$$\begin{pmatrix} \cdot & A' \\ \cdot & \cdot \end{pmatrix}$$

is also a block-encoding of  $A$ . One advantage of block-encoding is that for any quantum state  $|b\rangle$ , we can prepare  $A|b\rangle$  by considering  $U|0\rangle|b\rangle = \gamma|0\rangle \otimes A|b\rangle + |1\rangle(\dots)$ . Perform measurements on the first qubit, if the result is  $|0\rangle$ , then the post-selected quantum state is  $A|b\rangle/\|A|b\rangle\|_2$ .

Denote

$$U_{\mathcal{N}} = H^{\otimes m} \otimes I_2 \otimes I_M, \quad (5)$$

where  $H$  is the 2-by-2 Hadamard gate and  $I_M$  is the identity matrix of dimension  $M$ . Then for any  $i, j \in \{0, 1, \dots, M-1\}$ , we can check the validity of the following identity

$$\begin{aligned} & \langle i | \langle 0 |^{\otimes(m+1)} U_{\mathcal{M}}^\dagger U_{\mathcal{N}} | 0 \rangle^{\otimes(m+1)} | j \rangle \\ &= \left( \langle i | \otimes \left( \alpha \|A_i\|_2 \langle 0 | \langle A_i | + \langle 1 | (\dots) \right) \right) \left( \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle \otimes |0\rangle \otimes |j\rangle \right) \\ &= \frac{\alpha a_{ij}}{\sqrt{M}}. \end{aligned} \quad (6)$$

Therefore,  $U_{\mathcal{M}}^\dagger U_{\mathcal{N}}$  is a  $(\sqrt{M}/\alpha, m+1, 0)$  block-encoding of  $A$ . By definition,  $U_{\mathcal{M}}^\dagger U_{\mathcal{N}}$  is implemented in time  $O(m+T)$  in a quantum computer. Combing this block-encoding of  $A$  and lemma 9 of [23], we can construct a block-encoding of  $A^{-1}$ . The following lemma is a direct corollary of lemma 9 of [23] for the block-encoding  $U_{\mathcal{M}}^\dagger U_{\mathcal{N}}$  defined above.

**Lemma 1.** *Suppose that  $A$  is Hermitian. Let  $\kappa$  be the condition number of  $A$  and assume that  $I/\kappa \preceq A \preceq I$ . Then for any  $\epsilon$ , we can implement a  $(2\kappa, m+1 + \log(\kappa^2 \log \frac{1}{\epsilon}), \epsilon)$  block-encoding of  $A^{-1}$  in cost*

$$O\left(\kappa(m+T) \frac{\sqrt{M}}{\alpha} \log^2 \frac{\kappa^2}{\epsilon}\right). \quad (7)$$

Let  $x, y$  be two real unit vectors and  $|x\rangle, |y\rangle$  be their quantum states. When applying swap test, we can only obtain an estimation of  $|\langle x|y\rangle|$ . One simple technique to estimate  $\langle x|y\rangle$  is as follows [42], here  $\langle x|y\rangle$  should be understood as the inner product of  $x, y$  as real vectors: we first extend  $x, y$  into  $\tilde{x} = (1, x), \tilde{y} = (1, y)$ , then use swap test to estimate  $|\langle \tilde{x}|\tilde{y}\rangle|$ , which equals  $\frac{1}{2}(1 + \langle x|y\rangle)$ . Thus if we apply swap test to estimate  $|\langle \tilde{x}|\tilde{y}\rangle|$ , then we can get an estimation of  $\langle x|y\rangle$ . This simple technique can also be used to estimate  $\langle x|A|y\rangle$  for any matrix  $A$ . We just need to consider  $\langle \tilde{x}|\tilde{A}|\tilde{y}\rangle$ , where  $\tilde{A} := \text{diag}\{1, A\}$ . We will discuss this in more detail in the next section.

In machine learning, the data are usually real vectors. One goal of quantum machine learning is to find more efficient methods to solve machine learning problems by a quantum computer. From this point, we can assume that the input data of quantum machine learning are also real vectors. When using quantum computer to analyze these vectors, we first use some methods (e.g. qRAM) to get their quantum states. Based on the above analysis, we can consider the quantum state of  $\tilde{x}$  instead of  $x$  in quantum machine learning. This will not lose the generality to solve problems. This simple idea is widely used in neural networks [2]. Thus, in the following, when using swap test to estimate the inner product of quantum states, we always assume that the above simple technique is used. However, to simply the notations in this paper, we still use  $x$  instead of  $\tilde{x}$ .

**Proposition 1.** *Suppose that  $A$  is Hermitian. Let  $\kappa$  be the condition number of  $A$  and assume that  $I/\kappa \preceq A \preceq I$ . Let  $x, y$  be two real vectors such that their quantum states can be prepared in time  $O(T_x), O(T_y)$  respectively, then we can obtain an  $\epsilon$ -approximation of  $\langle x|A^{-1}|y\rangle$  in time*

$$O\left(\frac{\kappa}{\epsilon}\left(T_x + T_y + \kappa(m+T)\frac{\sqrt{M}}{\alpha}\log^2\frac{\kappa^2}{\epsilon}\right)\right). \quad (8)$$

**Proof.** By lemma 1 and remark 2, we can prepare  $\frac{1}{2\kappa}|0\rangle \otimes A^{-1}|y\rangle + |1\rangle(\cdots)$  in time

$$O\left(T_y + \kappa(m+T)\frac{\sqrt{M}}{\alpha}\log^2\frac{\kappa^2}{\epsilon}\right)$$

by using the block-encoding of  $A^{-1}$ . Then apply swap test to compute the inner product between  $\frac{1}{2\kappa}|0\rangle \otimes A^{-1}|y\rangle + |1\rangle(\cdots)$  and  $|0\rangle|x\rangle$ , we can compute an  $\epsilon'$ -approximation of  $\langle x|A^{-1}|y\rangle/2\kappa$  in time

$$O\left(\frac{1}{\epsilon'}\left(T_x + T_y + \kappa(m+T)\frac{\sqrt{M}}{\alpha}\log^2\frac{\kappa^2}{\epsilon}\right)\right).$$

The result (8) is obtained by setting  $\epsilon' = \epsilon/2\kappa$ .  $\square$

The above construction (1) and (5) of block-encoding is a generalization of the SVE method studied in [22]. In the SVE,  $U_{\mathcal{M}}, U_{\mathcal{N}}$  are defined by

$$U_{\mathcal{M}}|i, 0\rangle = |i\rangle|A_i\rangle \text{ and } U_{\mathcal{N}}|0, j\rangle = \frac{1}{\|A\|_F} \sum_{i=0}^{M-1} \|A_i\|_2 |i\rangle|j\rangle. \quad (9)$$

Then similar to the calculation of equation (6),  $U_{\mathcal{M}}^\dagger U_{\mathcal{N}}$  is a  $(\|A\|_F, m, 0)$ -block-encoding of  $A$ . The unitary operator  $U_{\mathcal{M}}$  in equation (9) is used to prepare the quantum states of rows of  $A$ . Generally, the preparation of quantum state is not easy for a quantum computer [15]. The SVE assumes the efficiency of qRAM [43] to prepare the quantum states of the given matrix.

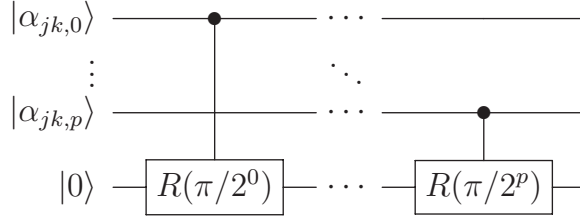
The fundamental study object of machine learning is data, and for quantum machine learning the data are quantum states. Thus, in the field of quantum machine learning, the qRAM assumption seems necessary nowadays. In this paper, we also need the assumption of qRAM to prepare the quantum states of the input data. However, in Bayes' classifiers, the entries of the matrices appeared in the discriminant functions are nonlinear functions of the input data. As a result, it is not straightforward to build the unitaries (9) for these matrices if we only have the quantum states of the input data. This is one reason why we modify SVE into the form (1) and (5).

The following is a simple idea to build  $U_{\mathcal{M}}$  defined in equation (1) if we have an efficient oracle to query the entries of the given matrix. Set  $\|A\|_{\max} = \max_{i,j} |a_{ij}|$ , define  $U_{\mathcal{M}}$  by

$$U_{\mathcal{M}}|i, 0\rangle = \frac{1}{\sqrt{M}\|A\|_{\max}} \sum_{j=0}^{M-1} a_{ij}|i, j\rangle|0\rangle + (\cdots)|1\rangle. \quad (10)$$

Assume that we have an efficient oracle  $\mathcal{O}$ , which is defined as  $\mathcal{O}|i, j\rangle|0\rangle = |i, j\rangle|a_{ij}\rangle$ , to query all the entries of  $A$ , then we can implement  $U_{\mathcal{M}}$  in time  $O(\log M)$ . More precisely, apply the oracle  $\mathcal{O}$  to the superposition  $\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |i, j\rangle|0\rangle$  to obtain  $\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |i, j\rangle|a_{ij}\rangle$ . Then apply control rotation based on the register  $|a_{ij}\rangle$  to prepare





**Figure 1.** The circuit of control rotation.

$$\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |i, j\rangle |a_{ij}\rangle \left( \frac{a_{ij}}{\|A\|_{\max}} |0\rangle + \sqrt{1 - \frac{|a_{ij}|^2}{\|A\|_{\max}^2}} |1\rangle \right). \quad (11)$$

Finally, we will get the desired result (10) by applying the inverse of the oracle to cancel the query of  $a_{ij}$  in the above state. The above simple idea has been used in [44, 45].

Even though  $M$  control rotations are used in generating the state (11), the complexity is independent of  $M$ . We can use one uniform quantum circuit to implement all the control rotations. More precisely, assume that in binary form  $\alpha_{ij} := \arccos(a_{ij}) \approx \sum_{k=0}^p \alpha_{ij,k} \pi / 2^k$  where  $\alpha_{ij,k} \in \{0, 1\}$ , then the rotation  $R(\alpha_{ij}) \approx \prod_k R(\alpha_{ij,k} \pi / 2^k) = \prod_{k: \alpha_{ij,k}=1} R(\pi / 2^k)$ . Thus to implement the control rotation, we first apply  $U_{\arccos} |x, 0\rangle = |x, \arccos(x)\rangle$  to generate  $\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |i, j\rangle |a_{ij}\rangle |\alpha_{ij,0}, \dots, \alpha_{ij,p}\rangle$ . With this state, we can implement control rotation by the circuit shown in figure 1. If we choose  $p = O(\log M)$ , then in this construction,  $O(T) = O(\log M)$  and  $\alpha = 1/\sqrt{M}\|A\|_{\max}$ .

Similar to the analysis of the SVE (see [22, theorem 11] or [23, theorem 27]), we can prove the following result. It is an application of quantum phase estimation.

**Proposition 2.** *Let  $A$  be an  $M \times M$  Hermitian matrix with eigenvalue decomposition  $A = \sum_{i=0}^{M-1} \sigma_i |u_i\rangle \langle u_i|$ . Then there is a quantum algorithm which runs in time  $O(\sqrt{M}(T' + T + \log^2 M)/\epsilon\alpha)$  to implement  $\sum_{i=0}^{M-1} \alpha_i |u_i\rangle |0\rangle \mapsto \sum_{i=0}^{M-1} \alpha_i |u_i\rangle |\tilde{\sigma}_i\rangle$ , where  $|\tilde{\sigma}_i - \sigma_i| \leq \epsilon$  for all  $i$ , and  $O(T')$  is the complexity to prepare  $\sum_{i=0}^{M-1} \alpha_i |u_i\rangle$ .*

**Remark 3.** Since we can choose simple initial state  $\sum_{i=0}^{M-1} \alpha_i |u_i\rangle$ , we now assume that  $T' = O(\log M)$ . With the choice of  $U_{\mathcal{M}}$  in equation (10), we can implement the quantum singular value decomposition (SVD) of general matrix in time  $\tilde{O}(M\|A\|_{\max}(\log M)^2/\epsilon)$  by proposition 2. In [31], with the same oracle assumption, Rebentrost *et al* obtained the quantum SVD of dense low-rank matrices based on Hamiltonian simulation technique. The cost is  $O(M\|A\|_{\max}^2(\text{poly log } M)/\epsilon^3)$ . Thus, compared to the algorithm of [31] the algorithm here has better dependence on  $\|A\|_{\max}$  and  $\epsilon$ .

Next, we consider how to estimate the determinant of positive definite Hermitian matrices. The following is a simple lemma to prove.

**Lemma 2.** *Assume that  $a_i, b_i > 0$  and  $|a_i - b_i| \leq \epsilon$  for  $i = 0, \dots, M-1$ , then*

$$\left| \sum_{i=0}^{M-1} \log a_i - \sum_{i=0}^{M-1} \log b_i \right| \leq \sum_{i=0}^{M-1} \frac{\epsilon}{b_i}. \quad (12)$$

**Proof.** Assume that  $a_i = b_i + \delta_i$ , then  $|\delta_i| \leq \epsilon$  and  $|\log a_i - \log b_i| = |\log(1 + \delta_i/b_i)| \approx |\delta_i|/b_i \leq \epsilon/b_i$ . Thus  $|\sum_{i=0}^{M-1} \log a_i - \sum_{i=0}^{M-1} \log b_i| \leq \sum_{i=0}^{M-1} |\log a_i - \log b_i| \leq \sum_{i=0}^{M-1} \epsilon/b_i$ .  $\square$

**Proposition 3.** Let  $A$  be an  $M \times M$  positive definite Hermitian matrix, then we can obtain an  $\epsilon$ -approximation of  $\log |A|$  in cost

$$O(\sqrt{M}\|A\|_F^2 \text{Tr}(A^{-1})(T_A + T + \log^2 M)(1 + |\log \sigma_{\max}|) |\log \sigma_{\min}| / \alpha \epsilon^2 \sigma_{\min}^2) \quad (13)$$

where  $O(T_A)$  is the complexity to prepare  $|A\rangle = \frac{1}{\|A\|_F} \sum_{i,j=0}^{M-1} a_{ij}|i,j\rangle$ , and  $\sigma_{\max}, \sigma_{\min}$  are respectively the maximal and minimal eigenvalues of  $A$ .

**Proof.** Assume that the eigenvalue decomposition of  $A$  is  $A = \sum_{i=0}^{M-1} \sigma_i |u_i\rangle\langle u_i|$ . By definition and the positive definite assumption,  $\log |A| = \sum_{i=0}^{M-1} \log \sigma_i$ .

By the definition of eigenvalue decomposition

$$|A\rangle = \frac{1}{\|A\|_F} \sum_{i=0}^{M-1} \sigma_i |u_i, u_i\rangle.$$

Due to proposition 2, we can obtain

$$\frac{1}{\|A\|_F} \sum_{i=0}^{M-1} \sigma_i |u_i, u_i\rangle |\tilde{\sigma}_i\rangle \quad (14)$$

in time  $O(\sqrt{M}(T_A + T + \log M)/\epsilon' \alpha)$ , where  $|\sigma_i - \tilde{\sigma}_i| \leq \epsilon'$ . Set

$$c = \min_{0 \leq i \leq M-1, \sigma_i \neq 1} \sigma_i / \sqrt{|\log \sigma_i|} = \sigma_{\min} / \sqrt{|\log \sigma_{\min}|},$$

and define

$$\delta(t) = \begin{cases} 0, & \text{if } t > 1; \\ 1, & \text{if } 0 < t \leq 1. \end{cases}$$

Apply control rotation to generate

$$\begin{aligned} & \frac{1}{\|A\|_F} \sum_{i=0}^{M-1} \sigma_i |u_i, u_i\rangle |\tilde{\sigma}_i\rangle |\delta(\tilde{\sigma}_i)\rangle \left( c \frac{\sqrt{|\log \tilde{\sigma}_i|}}{\tilde{\sigma}_i} |0\rangle + \sqrt{1 - \frac{c^2 |\log \tilde{\sigma}_i|}{\tilde{\sigma}_i^2}} |1\rangle \right) \\ &= \frac{c}{\|A\|_F} \sum_{i=0}^{M-1} \sqrt{|\log \tilde{\sigma}_i|} |u_i, u_i\rangle |\tilde{\sigma}_i\rangle |\delta(\tilde{\sigma}_i)\rangle |0\rangle + (\cdots) |1\rangle. \end{aligned}$$

The probability of the last two qubits equal  $|0\rangle|0\rangle$  is

$$P_0 = \frac{c^2}{\|A\|_F^2} \sum_{0 \leq i \leq M-1, \tilde{\sigma}_i > 1} \frac{\sigma_i^2}{\tilde{\sigma}_i^2} \log \tilde{\sigma}_i,$$

and the probability of the last two qubits equal  $|1\rangle|0\rangle$  is

$$P_1 = -\frac{c^2}{\|A\|_F^2} \sum_{0 \leq i \leq M-1, 0 < \tilde{\sigma}_i \leq 1} \frac{\sigma_i^2}{\tilde{\sigma}_i^2} \log \tilde{\sigma}_i.$$

Both of them can be approximated to precision  $\epsilon''$  by amplitude estimation method in cost

$$O(\sqrt{M}(T_A + T + \log M)/\epsilon' \epsilon'' \alpha).$$

Denote  $P = (P_0 - P_1)\|A\|_F^2/c^2 = \sum_{i=0}^{M-1} (\sigma_i^2/\tilde{\sigma}_i^2) \log \tilde{\sigma}_i$ , then

$$\begin{aligned} |P - \log |A|| &\leq \sum_{i=0}^{M-1} \frac{1}{\tilde{\sigma}_i^2} \left| \sigma_i^2 \log \tilde{\sigma}_i - \tilde{\sigma}_i^2 \log \sigma_i \right| \\ &\leq \sum_{i=0}^{M-1} \frac{1}{\tilde{\sigma}_i^2} \left( \tilde{\sigma}_i^2 |\log \tilde{\sigma}_i - \log \sigma_i| + |\log \tilde{\sigma}_i| |\sigma_i^2 - \tilde{\sigma}_i^2| \right) \\ &\leq \text{Tr}(A^{-1})\epsilon' + \sum_{i=0}^{M-1} \frac{\epsilon'}{\tilde{\sigma}_i^2} |\log \tilde{\sigma}_i| |\sigma_i + \tilde{\sigma}_i| \\ &\leq \text{Tr}(A^{-1})\epsilon' + \sum_{i=0}^{M-1} \frac{2\epsilon'}{\tilde{\sigma}_i} |\log \tilde{\sigma}_i| + \sum_{i=0}^{M-1} \frac{\epsilon'^2}{\tilde{\sigma}_i} |\log \tilde{\sigma}_i| \\ &\leq \text{Tr}(A^{-1})(1 + \epsilon' + 2 \max_i |\log \tilde{\sigma}_i|)\epsilon' \\ &\leq \text{Tr}(A^{-1})(2 + 2 \max_i |\log \tilde{\sigma}_i|)\epsilon'. \end{aligned}$$

The third inequality is due to lemma 2. In the fourth inequality, we use the fact that  $|\sigma_i| \leq |\tilde{\sigma}_i| + \epsilon$ . Apply the amplitude estimation, we obtain a  $P'$  such that  $|P - P'| \leq \|A\|_F^2 \epsilon''/c^2$ . Thus,

$$|P' - \log |A|| \leq \text{Tr}(A^{-1})(2 + 2 \max_i |\log \tilde{\sigma}_i|)\epsilon' + \frac{\|A\|_F^2}{c^2} \epsilon''.$$

Choose  $\epsilon', \epsilon''$  such that  $\text{Tr}(A^{-1})(2 + 2 \max_i |\log \tilde{\sigma}_i|)\epsilon' = \|A\|_F^2 \epsilon''/c^2 = \epsilon/2$ . With this choice,  $P'$  provides an  $\epsilon$ -approximation of  $\log |A|$ . The cost is (13) as claimed.  $\square$

Under the assumption of the SVE of [22], equation (13) becomes

$$O((\text{poly log } M)\|A\|_F^3 \text{Tr}(A^{-1})(1 + |\log \sigma_{\max}|)|\log \sigma_{\min}|/\epsilon^2 \sigma_{\min}^2). \quad (15)$$

If we can only obtain  $\frac{1}{\beta} \sum_{i=0}^{M-1} \|A_i\|_2 |i\rangle \langle A_i| |0\rangle + (\dots) |1\rangle$  in time  $O(T'_A)$ , then equation (13) reduces to

$$O(\sqrt{M} \beta^2 \text{Tr}(A^{-1})(T'_A + T + \log^2 M)(1 + |\log \sigma_{\max}|)|\log \sigma_{\min}|/\alpha \epsilon^2 \sigma_{\min}^2). \quad (16)$$

If we further assume that  $I/\kappa \preceq A \preceq I$ , then  $\text{Tr}(A^{-1}) \leq \kappa M$  and  $\|A\|_F \leq \sqrt{M}$ . So equations (15) and (16) can be respectively simplified into

$$O(\kappa^3 M^{2.5} (\text{poly log } M) (\log \kappa)/\epsilon^2) \quad (17)$$

and

$$O(\beta^2 \kappa^3 M^{1.5} (T'_A + T + \log^2 M) (\log \kappa)/\alpha \epsilon^2). \quad (18)$$

Another method to estimate  $\log |A|$  is as follows: in the quantum state (14), the amplitude of  $|\tilde{\sigma}_i\rangle$  is  $\sigma_i^2/\|A\|_F^2$ . Due to amplitude amplification technique, the cost to get  $\tilde{\sigma}_i$  by performing measurements is

$$O(\sqrt{M}\|A\|_F(T_A + T + \log M)/\epsilon'\alpha\sigma_i). \quad (19)$$

Thus, it costs

$$O(\sqrt{M}\|A\|_F\text{Tr}(A^{-1})(T_A + T + \log M)/\epsilon'\alpha) \quad (20)$$

to obtain all  $\tilde{\sigma}_i$ . By lemma 2,  $\sum_i \log \tilde{\sigma}_i$  is an  $\epsilon'/\text{Tr}(A^{-1})$ -approximation of  $\log |A|$ . To make this error small in size  $\epsilon$ , we choose  $\epsilon' = \epsilon\text{Tr}(A^{-1})$ . Thus the cost to get an  $\epsilon$ -approximation of  $\log |A|$  is

$$O(\sqrt{M}\|A\|_F\text{Tr}(A^{-1})^2(T_A + T + \log M)/\epsilon\alpha). \quad (21)$$

If  $\|A\|_F > \text{Tr}(A^{-1})$ , then the above method is better than the method proposed in proposition 3. However, under the assumption  $I/\kappa \preceq A \preceq I$ , equation (21) becomes  $O(\kappa^2 M^3(T_A + T + \log M)/\epsilon'\alpha)$ . Considering the dependence on  $M$ , it is worse than the classical method (which is at most  $O(M^3)$ ) and the method proposed in proposition 3.

### 3. Extract inner products of vectors in a quantum computer

Swap test [33] is a useful technique to estimate the inner product of two quantum states. It can be viewed as an application of amplitude estimation [46], and has been extensively studied in many literatures [34, 42, 47–49]. In this section, we first give a brief description of it. Then we use it to obtain some results that are useful in later sections.

Let

$$|\phi\rangle = \sin \theta |0\rangle|u\rangle + \cos \theta |1\rangle|v\rangle \quad (22)$$

be a quantum state that is prepared by a unitary operator  $U$  in time  $O(T_{\text{in}})$ . Amplitude estimation is a quantum algorithm that is used to estimate  $\pm\theta$  to accuracy  $\epsilon$  with high success probability. More precisely, let  $Z$  be the Pauli-Z matrix. Denote  $G = (I - 2|\phi\rangle\langle\phi|)(Z \otimes I)$ , then  $G$  is a rotation in the space spanned by  $\{|0\rangle|u\rangle, |1\rangle|v\rangle\}$ . The eigenvalues are  $e^{\pm i2\theta}$  and the corresponding eigenvectors are  $|w_{\pm}\rangle = \frac{1}{\sqrt{2}}(|0\rangle|u\rangle \pm i|1\rangle|v\rangle)$ .

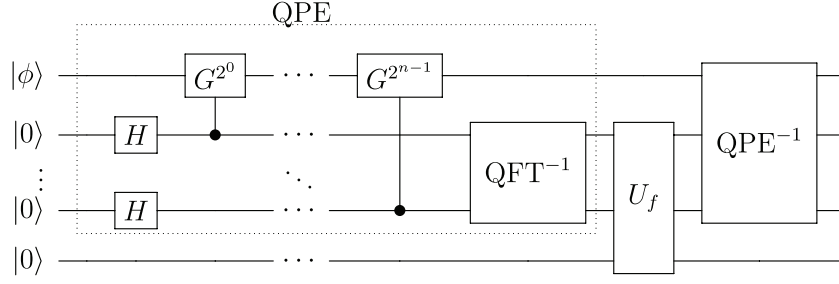
Rewrite the state  $|\phi\rangle$  as

$$|\phi\rangle = -\frac{i}{\sqrt{2}}(e^{i\theta}|w_+\rangle - e^{-i\theta}|w_-\rangle).$$

Then perform quantum phase estimation to  $G$  with the initial state  $|\phi\rangle|0\rangle^n$  for some  $n = O(\log 1/\delta\epsilon)$ . With probability at least  $1 - \delta$ , we will get an approximation of the following state

$$-\frac{i}{\sqrt{2}}(e^{i\theta}|w_+\rangle|y\rangle - e^{-i\theta}|w_-\rangle|y\rangle), \quad (23)$$

where  $y \in \mathbb{Z}_{2^n}$  satisfies  $|\theta - y\pi/2^n| \leq \epsilon$ . The time complexity of the above procedure is  $O(T_{\text{in}}/\epsilon\delta)$ . For more details about quantum phase estimation, we refer to [50, chapter 5]. Performing measurements on the state (23), we will get an  $\epsilon$ -approximation of  $\theta$  or  $-\theta$  with high probability at least  $1 - \delta$ . In the complexity analysis, we usually ignore  $\delta$  by setting it as a small constant.



**Figure 2.** Quantum circuit of amplitude estimation, where  $G = (I - 2|\phi\rangle\langle\phi|)(Z \otimes I)$ , and QFT refers to the quantum Fourier transform.

Furthermore, let  $f$  be an even function. Assume that there is an efficient oracle  $U_f$  to implement  $f$ , that is  $U_f|x, y\rangle = |x, y \oplus f(x)\rangle$  is an efficient unitary in the quantum computer. Then by equation (23), we can get

$$|\phi\rangle|f(\tilde{\theta})\rangle, \quad (24)$$

by adding a register to store  $f(\tilde{\theta})$  and undoing quantum phase estimation, where  $|\theta - \tilde{\theta}| \leq \epsilon$ . Since we perform no measurement to get equation (24), there is a unitary operator that maps  $|\phi\rangle|0\rangle$  to the state (24). To further apply this result in proposition 5, we draw its quantum circuit in figure 2.

**Remark 4.** Let  $a, b$  be two  $d$ -dimensional unit vectors, in this section, we always use the notation  $|\phi_{a,b}\rangle$  to represent the state

$$|\phi_{a,b}\rangle = \frac{1}{\sqrt{2}}(|+\rangle|a\rangle|b\rangle + |-\rangle|b\rangle|a\rangle), \quad (25)$$

where  $|\pm\rangle = (|0\rangle \pm |1\rangle)/2$ . We can prepare it by the quantum circuit shown in figure 3. If the cost to prepare  $|a\rangle, |b\rangle$  is  $O(T_{\text{in}})$ , then the cost to prepare  $|\phi_{a,b}\rangle$  is  $O(T_{\text{in}} + (\log d)^2)$ .

Now let  $x, y$  be two  $d$ -dimensional real unit vectors. Denote  $\tilde{x} = (1, x), \tilde{y} = (1, y)$ . Assume that  $|\tilde{x}\rangle, |\tilde{y}\rangle$  are prepared in time  $O(T_{\text{in}})$ . The amplitude estimation discussed above offers a quantum algorithm to estimate  $\langle x|y\rangle$ . To be more exact, consider the following quantum state

$$\begin{aligned} |\phi_{\tilde{x}, \tilde{y}}\rangle &= \frac{1}{\sqrt{2}}(|+\rangle|\tilde{x}\rangle|\tilde{y}\rangle + |-\rangle|\tilde{y}\rangle|\tilde{x}\rangle) \\ &= \frac{1}{2}|0\rangle(|\tilde{x}\rangle|\tilde{y}\rangle + |\tilde{y}\rangle|\tilde{x}\rangle) + \frac{1}{2}|1\rangle(|\tilde{x}\rangle|\tilde{y}\rangle - |\tilde{y}\rangle|\tilde{x}\rangle). \end{aligned} \quad (26)$$

Since  $|\tilde{x}\rangle, |\tilde{y}\rangle$  occupy  $O(\log d)$  qubits, the complexity to prepare  $|\phi_{\tilde{x}, \tilde{y}}\rangle$  is  $O(T_{\text{in}} + (\log d)^2)$ . Perform measurements on the first qubit of  $|\phi_{\tilde{x}, \tilde{y}}\rangle$ , then the probability to get  $|0\rangle$  is

$$\sin^2 \theta_1 = \frac{1}{2} \left( 1 + \frac{1}{4} (1 + \langle x|y\rangle)^2 \right). \quad (27)$$

By amplitude estimation, we can get an  $\epsilon'$ -approximation of  $\theta_1$ , denote as  $\tilde{\theta}_1$ . From  $\tilde{\theta}_1$  we can compute an estimation of  $\langle x|y\rangle$ . However, to get a good approximation of  $\langle x|y\rangle$  with small absolute error, it is not so good to use equation (27) directly because of the square term  $(1 + \langle x|y\rangle)^2$ . A much better way is as follows: denote  $\hat{y} = (-1, y)$ . If we consider the state  $|\phi_{\tilde{x}, \hat{y}}\rangle$ , then the probability to get  $|0\rangle$  is

$$\sin^2 \theta_2 = \frac{1}{2}(1 + |\langle \tilde{x} | \hat{y} \rangle|^2) = \frac{1}{2}(1 + \frac{1}{4}(1 - \langle x | y \rangle)^2). \quad (28)$$

Similarly, we can obtain an  $\epsilon'$ -approximation of  $\theta_2$ , denoted as  $\tilde{\theta}_2$ . Since  $2 \sin^2 \theta_1 - 2 \sin^2 \beta_2 = \langle x | y \rangle$  and  $|\sin^2 \theta_i - \sin^2 \tilde{\theta}_i| \leq 2\epsilon$  for  $i = 1, 2$ , we conclude that  $2 \sin^2 \tilde{\theta}_1 - 2 \sin^2 \tilde{\theta}_2$  provides a  $4\epsilon'$ -approximation of  $\langle x | y \rangle$ . Therefore,  $\langle x | y \rangle$  can be estimated in time  $O((T_{\text{in}} + (\log d)^2)/\epsilon)$  to precision  $\epsilon$  by setting  $\epsilon = 4\epsilon'$ .

Since the square of sine function is even, following the way to prepare (24), for any function  $f$  we can obtain

$$|\phi_{\tilde{x}, \tilde{y}}\rangle |\phi_{\tilde{x}, \tilde{y}}\rangle |f(s)\rangle, \quad (29)$$

where  $|s - \langle x | y \rangle| \leq \epsilon$ . Finally, we can summarize the above results into the following proposition.

**Proposition 4.** *Let  $x, y$  be two  $d$ -dimensional unit real vectors. Denote  $\tilde{x} = (1, x), \tilde{y} = (1, y), \hat{y} = (-1, y)$ . Assume that the quantum states  $|\tilde{x}\rangle, |\tilde{y}\rangle, |\hat{y}\rangle$  are prepared in time  $O(T_{\text{in}})$ . Let  $f$  be a univariate function such that  $U_f|x, y\rangle = |x, y \oplus f(x)\rangle$  is efficient. Then the following unitary transformation*

$$|\phi_{\tilde{x}, \tilde{y}}\rangle |\phi_{\tilde{x}, \tilde{y}}\rangle |0\rangle \mapsto |\phi_{\tilde{x}, \tilde{y}}\rangle |\phi_{\tilde{x}, \tilde{y}}\rangle |f(s)\rangle \quad (30)$$

can be achieved in time  $O((T_{\text{in}} + (\log d)^2)/\epsilon)$ , where  $|\langle x | y \rangle - s| \leq \epsilon$ .

As a corollary, we have the following result:

**Proposition 5.** *Let  $x_1, \dots, x_m, y_1, \dots, y_m$  be  $2m$   $d$ -dimensional unit real vectors. For any  $j = 1, \dots, m$ , denote  $\tilde{x}_j = (1, x_j), \tilde{y}_j = (1, y_j)$  and  $\hat{y}_j = (-1, y_j)$ . Assume that there is a unitary operator  $U$  that achieves the following transformation*

$$|j\rangle |0 \dots 0\rangle \mapsto |j\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \hat{y}_j}\rangle. \quad (31)$$

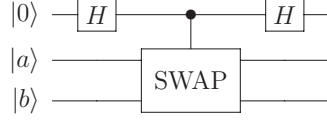
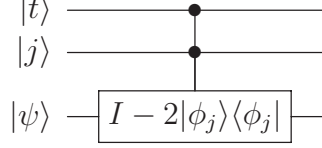
Then for any state  $\sum_{j=1}^m a_j |j\rangle$ , we can implement the following transformation

$$\sum_{j=1}^m a_j |j\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \hat{y}_j}\rangle |0\rangle \mapsto \sum_{j=1}^m a_j |j\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \hat{y}_j}\rangle |f(s_j)\rangle, \quad (32)$$

where  $|\langle x_j | y_j \rangle - s_j| \leq \epsilon$ . Moreover, if the complexity to implement the transformation (31) is  $O(T_U)$ , then the complexity to perform the transformation (32) is  $O((T_U + (\log d)^2)/\epsilon)$ .

**Proof.** The straightforward way to implement (32) is viewing  $|j\rangle$  as the control register to perform the transformation (30). The complexity of this control operation usually depends on  $m$ . However, due to the existence of the transformation (31), we can simplify this quantum circuit. For simplicity, set  $|\phi_j\rangle = |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \hat{y}_j}\rangle$ . Thus by equation (31),  $U|j, 0\rangle = |j, \phi_j\rangle$  for all  $j$ . By the circuit shown in figure 2, if we view  $|j\rangle$  as control register to apply the transformation (30) to  $|\phi_j\rangle$ , then we need to consider the implementation of the following control operator, where  $t \in \{0, 1\}$  and  $|\psi\rangle$  refers to the quantum state generated during the procedure (we do not need to know its expression precisely as the following analysis does not depend on it).

This is because the Hadamard gate, quantum Fourier transform and  $Z \otimes I$  are the same for all  $|\phi_j\rangle$ , there is no need to make a control to them by  $|j\rangle$ . In figure 4, the control of  $|t\rangle$  comes from the control operations in figure 2.

**Figure 3.** Quantum circuit to prepare  $|\phi_{a,b}\rangle$ .**Figure 4.** The circuit of control operations.

Note that

$$\begin{aligned}
 \sum_{j=1}^m |j\rangle\langle j| \otimes (I - 2|\phi_j\rangle\langle\phi_j|) &= I - 2 \sum_{j=1}^m |j, \phi_j\rangle\langle j, \phi_j| \\
 &= U \left( I - 2 \sum_{j=1}^m |j, 0\rangle\langle j, 0| \right) U^\dagger \\
 &= U \left( I \otimes (I - 2|0\rangle\langle 0|) \right) U^\dagger,
 \end{aligned}$$

so the circuit given in figure 4 is equivalent to the circuit of figure 5. In figure 5, the control of  $|j\rangle$  is removed.

Denote the unitary in figure 5 as  $\tilde{U}$ . Set  $|\Phi\rangle = \sum_{j=1}^m a_j |j\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle$ , then the quantum circuit to achieve the transformation (32) is the same as figure 2, where  $|\phi\rangle$  is changed into  $|\Phi\rangle$  and  $G$  is changed into  $\tilde{U}(Z \otimes I)$ . The complexity analysis comes naturally from the circuit.  $\square$

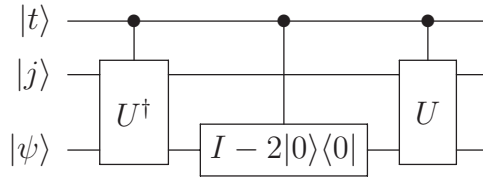
By storing  $\langle x_i | y_i \rangle$  in a register instead of measuring it can help us achieve many other goals efficiently. For instance, we can apply control rotation generated by  $f(s_i)$  to the result (32) to obtain

$$\sum_{j=1}^m a_j |j\rangle |\phi_j\rangle |f(s_j)\rangle \otimes \left( t f(s_j) |0\rangle + \sqrt{1 - |t f(s_j)|^2} |1\rangle \right), \quad (33)$$

where  $t = 1 / \max_j |f(s_j)|$ . Undo the procedure (32) and unprepare  $|\phi_j\rangle$ , then we get

$$\sum_{j=1}^m a_j |j\rangle \otimes \left( t f(s_j) |0\rangle + \sqrt{1 - |t f(s_j)|^2} |1\rangle \right). \quad (34)$$

As discussed in figure 1, the control rotation used in equation (33) is not expensive if we already the state  $\sum_{j=1}^m a_j |j\rangle |\phi_j\rangle |f(s_j)\rangle$ . By proposition 5, the complexity to get the state (34) is  $O((T_U + (\log d)^2)/\epsilon)$ . The procedure to obtain equation (34) is useful in constructing quantum naïve Bayes' classifier at the end of section 5.



**Figure 5.** The circuit of control operations: equivalent form.

**Remark 5.** The result of proposition 5 is frequently used in the section 5. However, the notation  $|\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle$  is pretty complicated. As discussed in section 2, in quantum machine learning, before preparing the quantum states of the input data  $x$ , we can intentionally change  $x$  into  $\tilde{x} = (1, x)$  in advance, then focus on the quantum state of  $\tilde{x}$ . Also, we can prepare  $|\tilde{y}\rangle$  from  $|\tilde{y}\rangle$ . So in the following, when using the result of proposition 5, we will simply use the state  $|\phi_{x,y}\rangle$  defined in remark 4 instead of  $|\phi_{\tilde{x}_j, \tilde{y}_j}\rangle |\phi_{\tilde{x}_j, \tilde{y}_j}\rangle$ .

#### 4. Bayes' classifiers

In this section, we give a review of the classical Bayes' classifiers. More details can be found in [1, 4]. Assume that there are  $q$  classes  $\mathcal{C}_1, \dots, \mathcal{C}_q$ . Let  $\mathbf{x}$  be a new data that need to be classified. The classification problem is to put  $\mathbf{x}$  into the right class based on certain criterions. Denote the *posterior probability* of the data  $\mathbf{x}$  belongs to the class  $\mathcal{C}_i$  as  $P(\mathcal{C}_i | \mathbf{x})$ , then by Bayes' theorem,

$$P(\mathcal{C}_i | \mathbf{x}) = \frac{P(\mathcal{C}_i)P(\mathbf{x} | \mathcal{C}_i)}{P(\mathbf{x})}, \quad (35)$$

where

- (i)  $P(\mathcal{C}_i)$  is the *prior probability*. It describes the probability of class  $\mathcal{C}_i$  in all classes. Thus  $\sum_i P(\mathcal{C}_i) = 1$ .
- (ii)  $P(\mathbf{x} | \mathcal{C}_i)$  is the *class likelihood*. It describes the conditional probability that a data belonging to  $\mathcal{C}_i$  has the associated observation value  $\mathbf{x}$ .
- (iii)  $P(\mathbf{x}) = \sum_i P(\mathbf{x} | \mathcal{C}_i)P(\mathcal{C}_i)$  is the *marginal probability*. It describes the probability that an observation  $\mathbf{x}$  is seen.

To minimize the error, the Bayes' classifier chooses the class of  $\mathbf{x}$  with the highest posterior probability. That is we put  $\mathbf{x}$  into the class  $\mathcal{C}_i$  if

$$P(\mathcal{C}_i | \mathbf{x}) = \max_{1 \leq j \leq q} P(\mathcal{C}_j | \mathbf{x}). \quad (36)$$

Since  $P(\mathbf{x})$  is the same for all  $P(\mathcal{C}_j | \mathbf{x})$ , we can define the *discriminant function* as

$$g_i(\mathbf{x}) = \log P(\mathcal{C}_i) + \log P(\mathbf{x} | \mathcal{C}_i). \quad (37)$$

A typical choice [51] of the conditional probability distribution  $P(\mathbf{x} | \mathcal{C}_i)$  is the Gauss distribution  $\mathcal{N}(\mu_i, \Sigma_i)$ , that is

$$P(\mathbf{x} | \mathcal{C}_i) = \frac{\exp \left[ -\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) \right]}{(2\pi)^{d/2} |\Sigma_i|^{1/2}}, \quad (38)$$



where  $\mu_i, \Sigma_i$  are respectively the mean and the covariance matrix of the samples associated with the class  $\mathcal{C}_i$ . With the choice of (38), we can change the discriminant function (37) into

$$g_i(\mathbf{x}) = -\frac{1}{2} \log |\Sigma_i| - \frac{1}{2} (\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) + \log P(\mathcal{C}_i), \quad (39)$$

where the constant term  $-\frac{d}{2} \log(2\pi)$  is ignored.

To apply supervised method to train the discriminant function (37) or (39), assume that we have a set of training samples  $\mathcal{X} = \{(\mathbf{x}^t, \mathbf{r}^t) : t = 0, \dots, N-1\}$ , where  $\mathbf{x}^t \in \mathbb{R}^d$ ,  $\mathbf{r}^t \in \{0, 1\}^q$  and  $\mathbf{r}_i^t = 1$  if  $\mathbf{x}^t \in \mathcal{C}_i$ . Since the samples are collected in advance, we assume that the number  $N_i := \sum_{t=0}^{N-1} \mathbf{r}_i^t$  of samples in the class  $\mathcal{C}_i$  is known. Therefore  $N = \sum_i N_i$ .

For any fixed  $i = 1, \dots, q$ , using maximum likelihood for each class, we define

$$\begin{aligned} P_i &= N_i/N, \\ \mathbf{m}_i &= \sum_{t=0}^{N-1} \mathbf{r}_i^t \mathbf{x}^t / N_i, \\ \mathbf{S}_i &= \sum_{t=0}^{N-1} \mathbf{r}_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T / N_i. \end{aligned} \quad (40)$$

It is easy to see that  $P_i$  counts the number of samples that lie in the  $i$ th class, which gives an approximation of  $P(\mathbf{x} | \mathcal{C}_i)$ . While  $\mathbf{m}_i$  and  $\mathbf{S}_i$  provide estimates of the corresponding mean  $\mu_i$  and covariance matrix  $\Sigma_i$ .

Substituting equations (40) into (39), the discriminant function becomes

$$g_i^{\text{quad}}(\mathbf{x}) = -\frac{1}{2} \log |\mathbf{S}_i| - \frac{1}{2} (\mathbf{x}^T \mathbf{S}_i^{-1} \mathbf{x} - 2\mathbf{x}^T \mathbf{S}_i^{-1} \mathbf{m}_i + \mathbf{m}_i^T \mathbf{S}_i^{-1} \mathbf{m}_i) + \log P_i. \quad (41)$$

This defines a *quadratic classifier*. Quadratic classifier can be simplified by pooling all the data in  $\mathcal{X}$ . Then we obtain another discriminant function defined by

$$g_i^{\text{linear}}(\mathbf{x}) = \mathbf{x}^T \mathbf{S}^{-1} \mathbf{m}_i - \frac{1}{2} \mathbf{m}_i^T \mathbf{S}^{-1} \mathbf{m}_i + \log P_i, \quad (42)$$

where

$$\mathbf{S} = \sum_{i=1}^q P_i \mathbf{S}_i = \frac{1}{N} \sum_{i=1}^q \sum_{t=0}^{N-1} \mathbf{r}_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T \quad (43)$$

is the common covariance matrix for all classes. In equation (42), we removed the common term  $-\frac{1}{2} \log |\mathbf{S}| - \frac{1}{2} \mathbf{x}^T \mathbf{S}^{-1} \mathbf{x}$ . Equation (42) defines a *linear classifier*.

Since linear classifier assumes that all the classes share a common covariance matrix, it has few parameters to estimate. Consequently, it is less flexible than the quadratic classifier, and so has substantially lower variance. This can potentially lead to improved performance on prediction. However, linear classifier can suffer from high bias if the assumption of a common covariance matrix for the classes is clearly untenable. Roughly, linear classifier tends to be better than quadratic classifier if there are relatively few training samples and thus reducing variance is crucial. In contrast, quadratic classifier is recommended if the training set is large so that the variance of the classifier is not a major concern.

We can further simplify the classifier (42) by assuming  $\mathbf{S} = \text{diag}\{s_1, \dots, s_d\}$  to be a diagonal matrix. This defines the *naïve Bayes' classifier*. It has the following simple form by adding the common term  $-\frac{1}{2} \mathbf{x}^T \mathbf{S}^{-1} \mathbf{x}$  into equation (42):

$$g_i^{\text{bayes}}(\mathbf{x}) = -\frac{1}{2} \sum_{j=1}^d \left( \frac{\mathbf{x}_j - \mathbf{m}_{ij}}{s_j} \right)^2 + \log P_i. \quad (44)$$

When  $s_1 = \dots = s_d = s$  in  $\mathbf{S}$ , then the classier function  $g_i^{\text{bayes}}$  defined in equation (44) can be further simplified into

$$g_i^{\text{mean}}(\mathbf{x}) = -\frac{1}{2s^2} \sum_{j=1}^d (\mathbf{x}_j - \mathbf{m}_{ij})^2 + \log P_i. \quad (45)$$

It reduces to evaluate  $\|\mathbf{x} - \mathbf{m}_i\|_2$ , which has been studied in [5]. If  $P_i = P_j$  for all  $i, j$ , then it is known as the *nearest mean classifier*. When assuming the qRAM to prepare the quantum states of all input samples efficiently, then paper [5] shows that quantum computer can complete this classification task in time  $O((\log Nd)/\epsilon)$ , which is exponential faster than the classical computer. However, as analyzed in [52], if the entries of the samples are relatively uniform (a case where state preparation is easy), then the classical random sampling algorithm solves the same problem in cost  $O((\log Nd)/\epsilon^2)$ , which is a little worse than the quantum algorithm.

In the context of near-term quantum technology, variational quantum circuit is an interesting model to design supervised quantum classifiers [11, 13, 53–56]. In this model, the unitaries on the quantum circuits contain parameters. Similar to the classical supervised machine learning algorithms, we can find suitable parameters by minimizing some loss functions. This kind of parametric methods is a little different from the one we studied in this paper. More precisely, in variational quantum circuit, the idea to introduce the parameters is similar to the weights used in neural networks. They play the role of adjusting the ability of the model. However, in Bayes's classifiers, we assume that there are certain statistical models among the samples. Usually, this is not necessary for the variational quantum circuits. From this point, quantum classifiers build on the variational quantum circuits can be viewed as non-parametric methods. Moreover, the parameters like the conditional probability in Bayes's classifiers can be estimated directly from the training samples without minimizing loss functions.

## 5. Quantum speedups of Bayes' classifiers

In this section, we show how to use quantum computer to speed up the above discussed three types of classifiers. For simplicity, we assume that  $\|\mathbf{x}^t\|_2 = 1$  for all  $t$ , and the quantum state  $|\mathbf{x}^t\rangle$  of  $\mathbf{x}^t$  are prepared efficiently such as by qRAM [43]. The first assumption is not necessary; however, it can simplify the notations below.

Under the qRAM assumption, each quantum state  $|\mathbf{x}^t\rangle$  can be prepared in  $O(\log^2 d)$  time steps. But the qRAM model requires  $O(\log d)$  address qubits,  $O(d)$  qutrits and  $O(d)$  classical or quantum memory cells. Moreover, it is not compatible with the quantum error-correction and fault-tolerant quantum computing. As shown in [57], the error rate per gate in this qRAM model is  $o(1/\sqrt{d})$  when it is used as an oracle for the quantum search problem. Recently in [49], Park *et al* proposed a standard circuit-based qRAM without using the routing algorithm. It can read classical data stored in memory cells, and superpose them in the computational basis states. In the following, we need to use  $|\mathbf{x}^t\rangle$  to prepare other quantum states, so we assume that  $\{|\mathbf{x}^t\rangle : t = 1, \dots, N\}$  are stored in memory cells so that they can be queried in superpositions efficiently.

### 5.1. Quantum speedup of quadratic classifier

In the following, we see how to achieve speedup at estimating  $g_i^{\text{quad}}(\mathbf{x})$  in a quantum computer. Denote

$$\mathcal{F}_i = \{t \in \{0, \dots, N-1\} : \mathbf{x}^t \in \mathcal{C}_i\}. \quad (46)$$

Then we can rewrite equation (40) as

$$\begin{aligned} \mathbf{m}_i &= \sum_{t \in \mathcal{F}_i} \mathbf{x}^t / N_i, \\ \mathbf{S}_i &= \sum_{t \in \mathcal{F}_i} (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T / N_i. \end{aligned} \quad (47)$$

**Remark 6.** To evaluate the quadratic discriminant (41), we need to compute  $\mathbf{S}_i^{-1}$ . It may happen that  $\mathbf{S}_i$  is not invertible. In this case, we can apply dimension reduction algorithm, such as principal component analysis, to obtain low dimensional vector representations of the training samples. Then perform the classification task in the low dimensional space. Another idea is to use the Moore–Penrose inverse. In the following, we assume that  $\mathbf{S}_i^{-1}$  is the Moore–Penrose inverse of  $\mathbf{S}_i$ . As for the determinant  $|\mathbf{S}_i|$ , it refers to the product of nonzero eigenvalues.

**Lemma 3.** *With the notations given in equation (47), we can prepare the following state in time  $O(\log^2 d + \log N_i)$*

$$|\tilde{\mathbf{m}}_i\rangle := \|\mathbf{m}_i\|_2 |0\rangle |\mathbf{m}_i\rangle + |1\rangle (\cdots). \quad (48)$$

**Proof.** We first prepare the superposition  $\frac{1}{\sqrt{N_i}} \sum_{t \in \mathcal{F}_i} |t\rangle$  by Hadamard operator. By the qRAM assumption, we can prepare  $\frac{1}{\sqrt{N_i}} \sum_{t \in \mathcal{F}_i} |t\rangle |\mathbf{x}^t\rangle$  by viewing  $|t\rangle$  as a control register. Finally, apply Hadamard operator again to  $|t\rangle$ , then we obtain  $\frac{1}{N_i} |0\rangle \sum_{t \in \mathcal{F}_i} |\mathbf{x}^t\rangle + |1\rangle (\cdots)$ . This gives the desired result.

The complexity of Hadamard operator is  $O(\log N_i)$ . By qRAM, the quantum state of each  $|\mathbf{x}^t\rangle$  can be prepared in time  $O(\log^2 d)$ . Therefore, the total cost of the above procedure is  $O(\log^2 d + \log N_i)$ .  $\square$

Denote the  $j$ th row of  $\mathbf{S}_i$  as  $\mathbf{S}_i(j)$ , then by equation (47)

$$\mathbf{S}_i(j) = \sum_{t \in \mathcal{F}_i} \frac{\mathbf{x}_j^t - \mathbf{m}_{ij}}{N_i} (\mathbf{x}^t - \mathbf{m}_i), \quad (49)$$

where  $\mathbf{x}_j^t, \mathbf{m}_{ij}$  are the  $j$ th component of  $\mathbf{x}^t$  and  $\mathbf{m}_i$  respectively.

**Lemma 4.** *For any  $j = 1, \dots, d$ , there is a unitary that achieves the transformation*

$$|j, 0\rangle \mapsto |j\rangle |\tilde{\mathbf{S}}_i(j)\rangle$$

*to precision  $\epsilon$  in cost  $O((\log^2 d + \log N_i)/\epsilon)$ , where*

$$|\tilde{\mathbf{S}}_i(j)\rangle = \frac{1}{4} \|\mathbf{S}_i(j)\|_2 |0\rangle |\mathbf{S}_i(j)\rangle + |1\rangle (\cdots).$$

**Proof.** For convenience, denote

$$\begin{aligned} |\psi_i^t\rangle &= \frac{1}{\sqrt{2}}(|0\rangle|0, \mathbf{x}^t\rangle - |1\rangle|\tilde{\mathbf{m}}_i\rangle), \\ |\phi_{ij}\rangle &= \frac{1}{\sqrt{2}}(|+\rangle|0, j\rangle|\tilde{\mathbf{m}}_i\rangle + |-\rangle|0, j\rangle|\tilde{\mathbf{m}}_i\rangle), \\ |\varphi_{ij}^t\rangle &= \frac{\mathbf{x}_j^t - \mathbf{m}_{ij}}{2}|0\rangle + \sqrt{1 - \frac{|\mathbf{x}_j^t - \mathbf{m}_{ij}|^2}{4}}|1\rangle. \end{aligned}$$

Since  $\|\mathbf{x}^t\|_2 = 1$  and  $\|\mathbf{m}_i\|_2 \leq 1$ , we have  $|\mathbf{x}_j^t - \mathbf{m}_{ij}| \leq 2$ . Thus  $|\varphi_{ij}^t\rangle$  is well-defined. Here, we simplified the notation  $|\phi_{ij}\rangle$  as stated in remark 5.

As analyzed in remark 4, we can prepare  $|\phi_{ij}\rangle$  from  $|0\rangle|0, j\rangle|\tilde{\mathbf{m}}_i\rangle$  by Hadamard gate and swap operator. Thus to apply the result of proposition 5, we need to construct a unitary to achieve

$$|j\rangle \otimes |0 \dots 0\rangle \mapsto |j\rangle \otimes |0\rangle|0, j\rangle|\tilde{\mathbf{m}}_i\rangle. \quad (50)$$

Lemma 3 states that there is a unitary to prepare  $|\tilde{\mathbf{m}}_i\rangle$  from  $|0\rangle$ . Since  $|\tilde{\mathbf{m}}_i\rangle$  is independent of  $j$  and it is easy to find a unitary to perform  $|j, 0\rangle \mapsto |j, j\rangle$ . Thus the unitary to implement (50) has complexity  $O(\log^2 d + \log N_i)$ .

By lemma 3 and the qRAM assumption, we can prepare  $|\psi_i^t\rangle$  in time  $O(\log^2 d + \log N_i)$ . As a result, we can obtain the following state in time  $O(\log^2 d + \log N_i)$  by viewing  $|t\rangle$  as a control qubit in the superposition  $\frac{1}{\sqrt{N_i}} \sum_{t \in \mathcal{F}_i} |t\rangle$

$$\frac{1}{\sqrt{N_i}} \sum_{t \in \mathcal{F}_i} |t\rangle \otimes |\psi_i^t\rangle \otimes |\phi_{ij}\rangle.$$

Note that  $\langle \tilde{\mathbf{m}}_i | 0, j \rangle = \mathbf{m}_{ij}$  and  $\mathbf{x}_j^t$  is known, thus by proposition 5 and equation (34) with  $f(s) = (\mathbf{x}_j^t - s)/2$ , we get

$$\frac{1}{\sqrt{N_i}} \sum_{t \in \mathcal{F}_i} |t\rangle \otimes |\psi_i^t\rangle \otimes |\phi_{ij}\rangle \otimes |\varphi_{ij}^t\rangle \quad (51)$$

in time  $O((\log^2 d + \log N_i)/\epsilon)$ . Undo the preparation of  $|\phi_{ij}\rangle$  to obtain

$$\frac{1}{\sqrt{N_i}} \sum_{t \in \mathcal{F}_i} |t\rangle \otimes |\psi_i^t\rangle \otimes |\varphi_{ij}^t\rangle.$$

Finally, apply Hadamard operator to  $|t\rangle$  and to the first register of  $|\psi_i^t\rangle$ , then after simplification, we get

$$\frac{1}{4N_i} |0\rangle \sum_{t \in \mathcal{F}_i} (\mathbf{x}_j^t - \mathbf{m}_{ij})(|\mathbf{x}^t\rangle - \|\mathbf{m}_i\|_2 |\mathbf{m}_i\rangle) + |1\rangle(\dots).$$

This gives the desired state. In the above state, we ignored the number, which is a constant, of qubits in  $|0\rangle$  of the first register.

Finally, we analyze the complexity of the above procedure. The main cost is to prepare (51). If the error to approximate  $\mathbf{S}_i(j)$  is controllable, then the complexity of the above procedure is  $O((\log^2 d + \log N_i)/\epsilon)$ . Thus it suffices to focus on the analysis of error caused by

swap test. Denote the  $\epsilon$ -approximation of  $\langle \tilde{\mathbf{m}}_i | 0, j \rangle = \mathbf{m}_{ij}$  obtained by swap test as  $\tilde{\mathbf{m}}_{ij}$ . Thus in practice,  $|\varphi_{ij}^t\rangle$  should be changed into

$$|\tilde{\varphi}_{ij}^t\rangle = \frac{\mathbf{x}_j^t - \tilde{\mathbf{m}}_{ij}}{2} |0\rangle + \sqrt{1 - \frac{|\mathbf{x}_j^t - \tilde{\mathbf{m}}_{ij}|^2}{4}} |1\rangle.$$

And the final state we actually obtain is

$$\frac{1}{4N_i} |0\rangle \sum_{t \in \mathcal{F}_i} (\mathbf{x}_j^t - \tilde{\mathbf{m}}_{ij}) (|\mathbf{x}^t\rangle - \|\mathbf{m}_i\|_2 |\mathbf{m}_i\rangle) + |1\rangle (\cdots)$$

as there is no error in preparing  $|\mathbf{m}_i\rangle$  by lemma 3. Denote

$$\hat{\mathbf{S}}_i(j) = \frac{1}{N_i} \sum_{t \in \mathcal{F}_i} (\mathbf{x}_j^t - \tilde{\mathbf{m}}_{ij}) (\mathbf{x}^t - \mathbf{m}_i),$$

then

$$\|\mathbf{S}_i(j) - \hat{\mathbf{S}}_i(j)\|_2 \leq \frac{1}{N_i} \sum_{t \in \mathcal{F}_i} |\mathbf{m}_{ij} - \tilde{\mathbf{m}}_{ij}| \|\mathbf{x}^t - \mathbf{m}_i\|_2 \leq 2\epsilon$$

in that  $\|\mathbf{x}^t\|_2, \|\mathbf{m}_i\|_2 \leq 1$ . Thus the error to approximate  $\mathbf{S}_i(j)$  is bounded by  $O(\epsilon)$ .  $\square$

The proof of lemma 4 provides an efficient approach to construct the unitary operator  $U_{\mathcal{M}}$  to prepare the quantum states of the rows of  $\mathbf{S}_i$ . In this case  $O(T) = O((\log^2 d + \log N_i)/\epsilon)$  and  $\alpha = 1/4$ . By lemma 4, we can prepare the following superposition of the rows of  $\mathbf{S}_i$

$$\frac{1}{\sqrt{d}} \sum_{j=1}^d |j\rangle |\tilde{\mathbf{S}}_i(j)\rangle = \frac{1}{4\sqrt{d}} \sum_{j=1}^d \|\mathbf{S}_i(j)\|_2 |j\rangle |\mathbf{S}_i(j)\rangle |0\rangle + (\cdots) |1\rangle \quad (52)$$

in cost  $O((\log^2 d + \log N_i)/\epsilon)$ . With the above preliminaries, now we can prove the following main result

**Theorem 1.** Suppose that  $\mathbf{x}$  is a new data need to be classified, then we can compute  $g_i^{\text{quad}}(\mathbf{x})$  to precision  $\epsilon$  in time

$$O\left(\frac{\kappa_i^2 \sqrt{d} (\log^2 d + \log N_i)}{\epsilon^2} \left(\frac{d^2 \log \kappa_i}{\epsilon} + \log^2 \frac{\kappa_i^2}{\epsilon}\right)\right) \quad (53)$$

in a quantum computer, where  $\kappa_i$  is the condition number of  $\mathbf{S}_i$ .

**Proof.** To simplify the notation, we assume that  $I/\kappa_i \preceq \mathbf{S}_i \preceq I$ . By lemma 4,  $O(T) = O((\log^2 d + \log N_i)/\epsilon)$  and  $\alpha = 1/4$ . By equation (52),  $T'_{\mathbf{S}_i} = O((\log^2 d + \log N_i)/\epsilon)$  and  $\beta = 1/4\sqrt{d}$ . Thus by equation (18), we can obtain an  $\epsilon$ -approximation of  $\log |\mathbf{S}_i|$  in time

$$O(\kappa_i^3 d^{2.5} (\log^2 d + \log N_i) (\log \kappa_i) / \epsilon^3).$$

By proposition 1 and lemma 3,  $\langle \mathbf{x} | \mathbf{S}_i^{-1} | \mathbf{x} \rangle$ ,  $\langle \mathbf{x} | \mathbf{S}_i^{-1} | \mathbf{m}_i \rangle$  and  $\langle \mathbf{m}_i | \mathbf{S}_i^{-1} | \mathbf{m}_i \rangle$  can be estimated to precision  $\epsilon$  in time

$$O\left(\frac{\kappa_i^2}{\epsilon^2}\sqrt{d}(\log^2 d + \log N_i) \log^2 \frac{\kappa_i^2}{\epsilon}\right).$$

Therefore, the final complexity to estimate  $g_i^{\text{quad}}(\mathbf{x})$  is (53).  $\square$

For a classical computer, computing the inverse and determinant of a  $d \times d$  matrix costs  $O(d^3)$  arithmetic operations [58]. It is easy to see from (40) and (41) that it costs  $O(N_i d^2 + d^3)$  to compute  $g_i^{\text{quad}}(\mathbf{x})$  in a classical computer, where  $O(N_i d^2)$  is used to calculate all the entries of  $\mathbf{S}_i$ . Compared to the classical algorithm, quantum computer achieves an exponential speedup at the number  $N_i$  of samples and a minor polynomial speedup at the dimension  $d$  when the condition number of  $\mathbf{S}_i$  is small.

### 5.2. Quantum speedup of linear classifier

The construction of the quantum algorithm to speed up linear classifier is similar to that of quantum quadratic classifier. Due to the simplified form, the quantum speedup at the dimension is better than quadratic classifier.

For each fixed  $t$ ,  $\mathbf{x}^t$  only belongs to one class, say  $\mathcal{C}_i$ , thus

$$\sum_{i=1}^q \mathbf{r}_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T = (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T. \quad (54)$$

Denote the  $j$ th row of  $\mathbf{S}$  as  $\mathbf{S}(j)$ , then by equations (43) and (54), we have

$$\mathbf{S}(j) = \sum_{t=0}^{N-1} \frac{\mathbf{x}_j^t - \mathbf{m}_{ij}}{N} (\mathbf{x}^t - \mathbf{m}_i),$$

where  $\mathbf{x}_j^t, \mathbf{m}_{ij}$  are the  $j$ th component of  $\mathbf{x}^t$  and  $\mathbf{m}_i$  respectively. It has a similar structure to equation (49). Hence, similar to the proof of lemma 4, we have

**Lemma 5.** *The following transformation can be prepared in time  $O((\log^2 d + \log N)/\epsilon)$*

$$|j\rangle|0\rangle \mapsto |j\rangle \left( \frac{1}{4} \|\mathbf{S}(j)\|_2 |0\rangle |\mathbf{S}(j)\rangle + |1\rangle (\cdots) \right). \quad (55)$$

As a result, we have

**Theorem 2.** *The value of  $g_i^{\text{linear}}$  at  $\mathbf{x}$  can be estimated to precision  $\epsilon$  in time*

$$O\left(\frac{\kappa^2 \sqrt{d}}{\epsilon^2} (\log^2 d + \log N) \log^2 \frac{\kappa^2}{\epsilon}\right) \quad (56)$$

*in a quantum computer, where  $\kappa$  is the condition number of  $\mathbf{S}$ .*

**Proof.** Compared to quadratic classifier, linear classifier does not need to estimate  $\log |\mathbf{S}|$ . So the proof of this theorem is the second part of that of theorem 1. Substitute the result of lemmas 3 and 5 into proposition 1, we conclude that  $\langle \mathbf{x} | \mathbf{S}_i^{-1} | \mathbf{x} \rangle, \langle \mathbf{m}_i | \mathbf{S}_i^{-1} | \mathbf{m}_i \rangle$  can be estimated to precision  $\epsilon$  in time (56).  $\square$

Similar to the analysis of quadratic classifier, the classical method to evaluate  $g_i^{\text{linear}}(\mathbf{x})$  costs  $O(N_id^2 + d^3)$ . When the condition number of  $\mathbf{S}$  is small, then quantum linear classifier achieves exponential speedup at the number of samples and sextic speedup at the dimension over the classical counterpart.

### 5.3. Quantum speedup of naïve Bayes' classifier

In this last subsection, we consider the quantum speedup of naïve Bayes' classifier. The quantum speedup here only depends on the swap test discussed in section 3.

By equation (43), we have

$$s_j = \frac{1}{N} \sum_{t=0}^{N-1} (\mathbf{x}_j^t - \mathbf{m}_{ij})^2, \quad (57)$$

where  $\mathbf{x}^t \in \mathcal{C}_i$ .

In a classical computer, calculating each  $s_j$  costs  $O(N)$  operations. In equation (44), there are  $d$  terms, thus the total cost of the classical method to accomplish naïve Bayes' classifier is  $O(Nd)$ . However, the following result shows that quantum computer can do exponentially better.

**Theorem 3.** *Let  $\mathbf{x}$  be a new data, then we can estimate  $g_i^{\text{bayes}}(\mathbf{x})$  to precision  $\epsilon$  in time  $O((\log^2 d + \log N)/\epsilon^3)$  in a quantum computer.*

**Proof.** First, we show how to compute  $s_j$ . By lemma 3,  $|\tilde{\mathbf{m}}_i\rangle$  can be prepared in time  $O(\log^2 d + \log N_i) = O(\log^2 d + \log N)$ . By viewing  $|t\rangle$  as a control qubit in  $\frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |t\rangle$ , we can prepare the following state in time  $O(\log^2 d + \log N)$ .

$$\frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |t\rangle \otimes \frac{1}{\sqrt{2}} \left( |+\rangle |0, j\rangle |\tilde{\mathbf{m}}_i\rangle + |-\rangle |0, j\rangle |\tilde{\mathbf{m}}_i\rangle \right).$$

Since  $|0, j\rangle$  is independent of the summation in the above state, to apply proposition 5 we need to find a unitary to achieve

$$|t, 0\rangle \mapsto |t, \tilde{\mathbf{m}}_i\rangle. \quad (58)$$

This is the same as implementing (50) in the proof of lemma 4. More precisely, by lemma 3 in preparing  $|\tilde{\mathbf{m}}_i\rangle$ , except the Hadamard gate, the other unitaries are the ones used in qRAM. Based on the assumption of qRAM, we assume that  $|t, 0\rangle \mapsto |t, \mathbf{x}^t\rangle$  is obtainable. Thus, we have a unitary to perform the transformation (58).

Now by proposition 5 and equation (33), we can further prepare

$$\begin{aligned} & \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |t\rangle \otimes \frac{1}{\sqrt{2}} \left( |+\rangle |0, j\rangle |\tilde{\mathbf{m}}_i\rangle + |-\rangle |0, j\rangle |\tilde{\mathbf{m}}_i\rangle \right) \\ & \otimes \left( \frac{(\mathbf{x}_j^t - \mathbf{m}_{ij})^2}{4} |0\rangle - \sqrt{1 - \frac{(\mathbf{x}_j^t - \mathbf{m}_{ij})^4}{16}} |1\rangle \right) \end{aligned}$$

in time  $O((\log^2 d + \log N)/\epsilon)$  by choosing  $f(s) = (\mathbf{x}_j^t - s)^2/4$ . Here  $\epsilon$  is the precision to approximate  $\mathbf{m}_{ij}$ . Unprepare  $\frac{1}{\sqrt{2}}(|+\rangle |0, j\rangle |\tilde{\mathbf{m}}_i\rangle + |-\rangle |0, j\rangle |\tilde{\mathbf{m}}_i\rangle)$ , and apply Hadamard operator

to  $|t\rangle$  to get

$$|\tilde{s}_j\rangle := \frac{1}{4N} \sum_{t=0}^{N-1} (\mathbf{x}_j^t - \mathbf{m}_{ij})^2 |0\rangle + |0\rangle^\perp = \frac{s_j}{4} |0\rangle^{\otimes \lceil \log N \rceil} |0\rangle + |0\rangle^\perp.$$

Now,  $s_j$  can be obtained by applying amplitude estimation to estimate the amplitude of  $|0\rangle$  of  $|\tilde{s}_j\rangle$ .

In the following, we consider the error caused by amplitude estimation. Denote the  $\epsilon$ -approximation of  $\mathbf{m}_{ij}$  as  $\mathbf{m}'_{ij}$ , then the amplitude of  $|0\rangle^{\otimes \lceil \log N \rceil} |0\rangle$  of  $|\tilde{s}_j\rangle$  is

$$\frac{s'_j}{4} := \frac{1}{4N} \sum_{t=0}^{N-1} (\mathbf{x}_j^t - \mathbf{m}'_{ij})^2.$$

Since  $\|\mathbf{x}'\|_2 = 1$  and  $\|\mathbf{m}_i\|_2 \leq 1$ , we have

$$\frac{1}{4} |s_j - s'_j| \leq \frac{1}{4N} \sum_{t=0}^{N-1} |\mathbf{m}_{ij} - \mathbf{m}'_{ij}| |2\mathbf{x}_j^t - \mathbf{m}_{ij} - \mathbf{m}'_{ij}| \leq \epsilon.$$

Thus the error to approximate  $s_j$  is bounded by  $4\epsilon$ , which is controllable.

When considering about equation (44), the quantum state  $|\tilde{s}_j\rangle$  will play a role to estimate  $g_i^{\text{bayes}}(\mathbf{x})$ . To be more precise, viewing  $|+\rangle, |-\rangle$  as control registers we can prepare

$$|\phi_0\rangle = \frac{1}{\sqrt{2}} (|+\rangle|0\rangle|\mathbf{x}\rangle - |-\rangle|\tilde{\mathbf{m}}_i\rangle) = \frac{1}{2} |0\rangle \sum_{j=1}^d (\mathbf{x}_j - \mathbf{m}_{ij}) |j\rangle + |0\rangle^\perp$$

in time  $O(\log^2 d + \log N)$  due to lemma 3. Viewing  $|0, j\rangle$  as control qubits in  $|\phi_0\rangle$ , we obtain

$$\frac{1}{2} |0\rangle \sum_{j=1}^d (\mathbf{x}_j - \mathbf{m}_{ij}) |j\rangle \otimes \frac{1}{\sqrt{2}} (|+\rangle|\tilde{s}_j\rangle + |-\rangle|0\rangle^{\otimes \lceil \log N \rceil} |0\rangle) + |0\rangle^\perp$$

in time  $O((\log^2 d + \log N)/\epsilon)$ . Set  $c = \min_j |s_j|$ , then by proposition 5 with  $f(s) = 4s/c$ , we can generate

$$\begin{aligned} & \frac{1}{2} |0\rangle \sum_{j=1}^d (\mathbf{x}_j - \mathbf{m}_{ij}) |j\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle|\tilde{s}_j\rangle + |1\rangle|0\rangle^{\otimes \lceil \log N \rceil} |0\rangle) \\ & \otimes \left( \frac{c}{s_j} |0\rangle + \sqrt{1 - \left| \frac{c}{s_j} \right|^2} |1\rangle \right) + |0\rangle^\perp \end{aligned}$$

in time  $O((\log^2 d + \log N)/\epsilon^2)$ . Now unprepare  $\frac{1}{\sqrt{2}} (|0\rangle|\tilde{s}_j\rangle + |1\rangle|0\rangle^{\otimes \lceil \log N \rceil} |0\rangle)$  returns

$$\begin{aligned} & \frac{1}{2} |0\rangle \sum_{j=1}^d (\mathbf{x}_j - \mathbf{m}_{ij}) |j\rangle \left( \frac{c}{s_j} |0\rangle + \sqrt{1 - \left| \frac{c}{s_j} \right|^2} |1\rangle \right) + |0\rangle^\perp \\ & = \frac{c}{2} |0\rangle \sum_{j=1}^d \frac{\mathbf{x}_j - \mathbf{m}_{ij}}{s_j} |j\rangle |0\rangle + |0\rangle^\perp. \end{aligned} \tag{59}$$

Apply swap operations to (59), we can transform  $|j\rangle|0\rangle$  into  $|0\rangle|j\rangle$ , thus we have



**Table 1.** Comparison of the efficiency of classical and quantum classifiers, where  $N$  is the number of samples,  $d$  is the dimension,  $\tilde{\kappa} = \max_i \kappa_i$ ,  $\kappa_i$  is the condition number of  $\mathbf{S}_i$ ,  $\kappa$  is the condition number of  $\mathbf{S}$ , and  $\epsilon$  is the precision. To simplify the notations, we use  $\tilde{O}$  to omit the lower terms.

	Classical algorithm	Quantum algorithm
Quadratic classifier	$O(Nd^2 + d^3)$	$\tilde{O}(\tilde{\kappa}^2 d^{2.5} (\log^2 d + \log N)/\epsilon^3)$
Linear classifier	$O(Nd^2 + d^3)$	$\tilde{O}(\kappa^2 d^{0.5} (\log^2 d + \log N)/\epsilon^2)$
Naïve Bayes' classifier	$O(Nd)$	$O((\log^2 d + \log N)/\epsilon^3)$

$$|\phi_1\rangle := \frac{c}{2}|0,0\rangle \sum_{j=1}^d \frac{\mathbf{x}_j - \mathbf{m}_{ij}}{s_j} |j\rangle + |0,0\rangle^\perp.$$

Note that the first term of  $g_i^{\text{bayes}}(\mathbf{x})$  can be approximated by computing the inner product between the first terms of  $|\phi_0\rangle$  and  $|\phi_1\rangle$ . To compute their inner product, we add an ancilla qubit into  $|\phi_0\rangle$  and  $|\phi_1\rangle$ , such that they become

$$\begin{aligned} |\tilde{\phi}_0\rangle &= \frac{1}{2}|0\rangle|0,0\rangle \sum_{j=1}^d (\mathbf{x}_j - \mathbf{m}_{ij})|j\rangle + |1\rangle|0,0\rangle^\perp, \\ |\tilde{\phi}_1\rangle &= \frac{c}{2}|0\rangle|0,0\rangle \sum_{j=1}^d \frac{\mathbf{x}_j - \mathbf{m}_{ij}}{s_j} |j\rangle + |2\rangle|0,0\rangle^\perp. \end{aligned}$$

This is achievable by control operations since the second term in  $|\phi_0\rangle$  or  $|\phi_1\rangle$  is orthogonal to the first term.

Now apply swap test to compute the inner product between  $|\tilde{\phi}_0\rangle$  and  $|\tilde{\phi}_1\rangle$ , we get an estimation of the first term of  $g_i^{\text{bayes}}(\mathbf{x})$ . The second term of  $g_i^{\text{bayes}}(\mathbf{x})$  is easy to compute by equation (40).

Based on the above construction, the costs to prepare  $|\tilde{\phi}_0\rangle$  and  $|\tilde{\phi}_1\rangle$  are  $O((\log^2 d + \log N)/\epsilon)$  and  $O((\log^2 d + \log N)/\epsilon^2)$ , respectively. Therefore, performing swap test to precision  $\epsilon$  to estimate  $g_i^{\text{bayes}}(\mathbf{x})$  costs  $O((\log^2 d + \log N)/\epsilon^3)$ .  $\square$

Finally, we summarize the results obtained in this section in table 1.

## 6. Conclusions

In this paper, we applied quantum algorithms to speed up three Bayes' classifiers (quadratic, linear and naïve Bayes). Under the assumption of qRAM to prepare the quantum states of training samples, polynomial and exponential speedup are obtained. The main technique we used are block-encoding and swap test, which are powerful methods to handle matrix-vector operations in the quantum computer. From the definition and the theory constructed in [23], block-encoding is more suitable to manipulate matrix operations when the matrix is given directly; i.e. we know all the entries explicitly. However, the matrices in many machine learning problems, such as Laplace eigenmap [59], Fisher discriminant analysis [60], and the problems considered in this paper, are not belonging to this case. Usually, extra calculations are required. For instance, in Laplace eigenmap, the entries of the Hermitian matrix is the Gauss

function of the input samples. In Fisher discriminant analysis, the matrix is a summation of rank-one matrices, which are generated by the input samples and the means of the samples in different classes. So to be more practical, we need to find better approaches to construct the block-encoding of the given matrix. Some approaches are proposed in [37]. The method proposed in this paper is also a simple method to find block-encoding that may have other applications.

## Acknowledgment

This work was supported by the QuantERA ERA-NET Cofund in Quantum Technologies implemented within the European Union's Horizon 2020 Programme (QuantAlgo project), and EPSRC Grants EP/L021005/1 and EP/R043957/1. No new data were created during this study.

## ORCID iDs

Changpeng Shao  <https://orcid.org/0000-0002-3008-7296>

## References

- [1] Alpaydin E 2015 *Introduction to Machine Learning* 3rd edn (Cambridge, MA: MIT Press)
- [2] Haykin S 2009 *Neural Networks and Learning Machines* 3rd edn (London: Pearson)
- [3] Hastie T, Tibshirani R and Friedman J 2016 *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 2nd edn (New York: Springer)
- [4] James G, Witten D, Hastie T and Tibshirani R 2013 *An Introduction to Statistical Learning: With Applications in R* (New York: Springer)
- [5] Lloyd S, Mohseni M and Rebentrost P 2013 Quantum algorithms for supervised and unsupervised machine learning (arXiv:1307.0411v2)
- [6] Wiebe N, Kapoor A and Svore K 2015 Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning *Quantum Inf. Comput.* **15** 0316–56
- [7] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503
- [8] Aïmeur E, Brassard G and Gambs S 2007 Quantum clustering algorithms *Proc. 24th Annual Int. Conf. of Machine Learning* pp 1–8
- [9] Cong I and Duan L 2016 Quantum discriminant analysis for dimensionality reduction and classification *New J. Phys.* **18** 073011
- [10] Schuld M, Sinayskiy I and Petruccione F 2014 The quest for a quantum neural network *Quantum Inf. Process* **13** 2567–86
- [11] Farhi E and Neven H 2018 Classification with quantum neural networks on near term processors (arXiv:1802.06002)
- [12] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504
- [13] Havlicek V, Corcoles A, Temme K, Harrow A, Kandala A, Chow J and Gambetta J 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
- [14] Schuld M, Sinayskiy I and Petruccione F 2014 An introduction to quantum machine learning *Contemp. Phys.* **56** 172–85
- [15] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
- [16] Maron M E 1961 Automatic indexing: an experimental inquiry *J. ACM* **8** 404–17
- [17] Rish I 2001 An empirical study of the naive Bayes classifier *Proc. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence* vol 3 pp 41–6

- [18] McCallum A and Nigam K 1998 A comparison of event models for naive bayes text classification *AAAI-98 Workshop on Learning for Text Categorization* vol 752 pp 41–8
- [19] Gelman A, Carlin J, Stern H, Dunson D, Vehtari A and Rubin D 2013 *Bayesian Data Analysis* 3rd edn (London: Chapman and Hall)
- [20] Murphy K and Bach F 2012 *Machine Learning: a Probabilistic Perspective* (Cambridge, MA: MIT Press)
- [21] Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502
- [22] Kerenidis I and Prakash A 2017 Quantum recommendation systems *Proc. 8th Innovations in Theoretical Computer Science Conf.* vol 67 pp 1–21
- [23] Chakraborty S, Gilyén A and Jeffery S 2019 The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation *Proc. 46th International Colloquium on Automata, Languages, and Programming* vol 132 pp 1–14
- [24] Kerenidis I, Landman J, Luongo A and Prakash A 2018  $q$ -means: a quantum algorithm for unsupervised machine learning (arXiv:1812.03584v2)
- [25] Kerenidis I and Luongo A 2018 Quantum classification of the MNIST dataset via slow feature analysis (arXiv:1805.08837v2)
- [26] Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3
- [27] Wiebe N, Braun D and Lloyd S 2012 Quantum algorithm for data fitting *Phys. Rev. Lett.* **109** 050505
- [28] Zhao Z, Fitzsimons J K and Fitzsimons J F 2019 Quantum assisted Gaussian process regression *Phys. Rev. A* **99** 052331
- [29] Zhao Z, Pozas-Kerstjens A, Rebentrost P and Wittek P 2019 Bayesian deep learning on a quantum computer *Quantum Mach. Intell.* **1** 41–51
- [30] Berry D W, Childs A M and Kothari R 2015 Hamiltonian simulation with nearly optimal dependence on all parameters *Proc. 56th IEEE Symp. on Foundations of Computer Science* pp 792–809
- [31] Rebentrost P, Steffens A, Marvian I and Lloyd S 2018 Quantum singular value decomposition of non-sparse low-rank matrices *Phys. Rev. A* **97** 012327
- [32] Wang C H and Wossnig L 2018 A quantum algorithm for simulating non-sparse Hamiltonians (arXiv:1803.08273)
- [33] Buhrman H, Cleve R, Watrous J and de Wolf R 2001 Quantum fingerprinting *Phys. Rev. Lett.* **87** 167902
- [34] Shao C 2018 A quantum model for multilayer perceptron (arXiv:1808.10561)
- [35] Schuld M, Fingerhuth M and Petruccione F 2017 Implementing a distance-based classifier with a quantum interference circuit *Europhys. Lett.* **119** 60002
- [36] Shao C and Xiang H 2018 Quantum regularized least squares solver with parameter estimate (arXiv:1812.09934)
- [37] Gilyén A, Su Y, Low G H and Wiebe N 2019 Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics *Proc. 51st Annual ACM SIGACT Symposium on Theory of Computing* pp 193–204
- [38] Low G and Chuang I 2019 Hamiltonian simulation by qubitization *Quantum* **3** 163
- [39] Kerenidis I and Prakash A 2018 A quantum interior point method for LPs and SDPs (arXiv:1808.09266)
- [40] van Apeldoorn J and Gilyén A 2018 Improvements in quantum SDP-solving with applications (arXiv:1804.05058)
- [41] Shao C and Xiang H 2019 Randomized row and column iterative methods with a quantum computer (arXiv:1905.11686)
- [42] Schuld M and Petruccione F 2018 *Supervised Learning With Quantum Computers* (Switzerland: Springer)
- [43] Giovannetti V, Lloyd S and Maccone L 2008 Quantum random access memory *Phys. Rev. Lett.* **100** 160501
- [44] Li T, Chakrabarti S and Wu X 2019 Sublinear quantum algorithms for training linear and Kernel-based classifiers *Proc. of the 36th Int. Conf. on Machine Learning* vol PMLR 97 pp 3815–24
- [45] Clader B, Jacobs B and Sprouse C 2013 Preconditioned quantum linear system algorithm *Phys. Rev. Lett.* **110** 250504
- [46] Brassard G, Høyer P and Mosca M 2002 Quantum amplitude amplification and estimation (*Quantum Computation and Information: A Millennium Volume*) *Contemp. Math.* **305** 53–74

- [47] Shao C, Li Y and Li H 2019 Quantum algorithm design: techniques and applications *J. Syst. Sci. Complex* **32** 375–452
- [48] Schuld M, Sinayskiy I and Petruccione F 2016 Prediction by linear regression on a quantum computer *Phys. Rev. A* **94** 022342
- [49] Park D, Petruccione F and Rhee J 2019 Circuit-based quantum random access memory for classical data *Sci. Rep.* **9** 3949
- [50] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge: Cambridge University Press)
- [51] Duda R O, Hart P E and Stork D G 2001 *Pattern Classification* 2nd edn (New York: Wiley)
- [52] Aaronson S 2015 Quantum machine learning algorithms: read the fine print *Nat. Phys.* **11** 291–3
- [53] Schuld M, Bocharov A, Svore K and Wiebe N 2018 Circuit-centric quantum classifiers (arXiv:1804.00633)
- [54] Coyle B, Mills D, Danos V and Kashefi E 2019 The born supremacy: quantum advantage and training of an Ising Born machine (arXiv:1904.02214v2)
- [55] Shao C 2019 A simple approach to design quantum neural networks and its applications to Kernel-learning methods (arXiv:1910.08798)
- [56] Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green A and Severini S 2018 Hierarchical quantum classifiers *npj Quantum Inf.* **4** 65
- [57] Arunachalam S, Gheorghiu V, Jochym-O'Connor T, Mosca M and Srinivasan P 2015 On the robustness of bucket brigade quantum RAM *New J. Phys.* **17** 123010
- [58] Golub H H and Van Loan C F 2013 *Matrix Computations* 4th edn (Baltimore, MD: John Hopkins University Press)
- [59] Belkin M and Niyogi P 2003 Laplacian eigenmaps for dimensionality reduction and data representation *Neural Comput.* **15** 1373–96
- [60] McLachlan G 2004 *Discriminant Analysis and Statistical Pattern Recognition* (New York: Wiley)